6lo Fragmentation DT

Announcement and Problem Statement

Thomas Watteyne (Chair)
Carsten Bormann
Rahul Jadhav
Gorry Fairhurst
Pascal Thubert
Gabriel Montenegro

Outline

- Per-hop reassembly solutions
  - RFC4944
  - RFC6282
- Problem statement
- Fragment forwarding solutions
  - Carsten’s book
  - draft-bormann-lwig-6lowpan-virtual-reassembly-00
  - draft-watteyne-6lo-minimal-fragment *(unpublished)*
  - draft-thubert-6lo-forwarding-fragments-08
- Use cases and open discussion

RFC4944

- Link-layer fragmentation only in route-over → reassembly at ea op
- Fragment header

Units of 8 octets

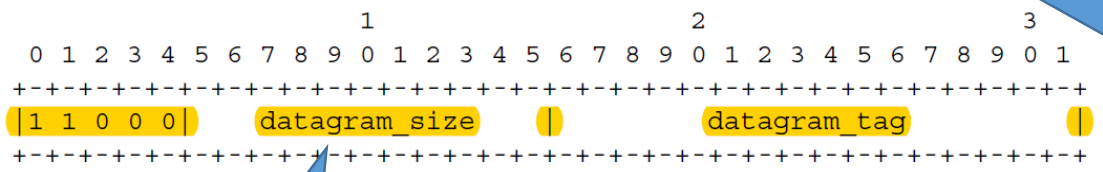


Figure 4: First Fragment

bytes

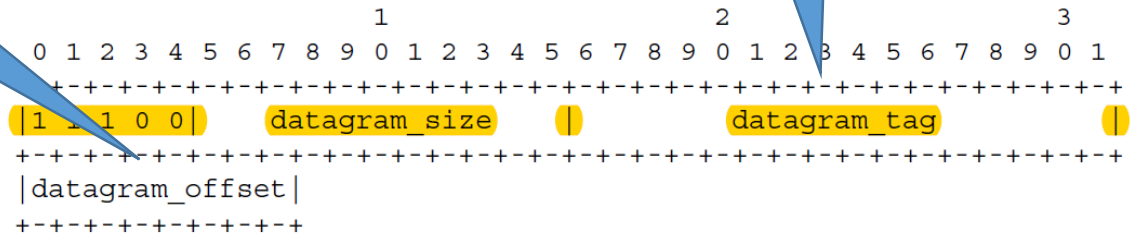


Figure 5: Subsequent Fragments

- +1 on each new frag
- No initial value specified

- Reassembly timer:
  - Starts when node receives first fragment
  - Timeout value MUST be <60s
  - When times out, buffer cleared, packet dropped

Pattern	Header Type	
00 xxxxxx	NALP	- Not a LoWPAN frame
01 000001	IPv6	- Uncompressed IPv6 Addresses
01 000010	LOWPAN_HC1	- LOWPAN_HC1 compressed IPv6
01 000011	reserved	- Reserved for future use
...	reserved	- Reserved for future use
01 001111	reserved	- Reserved for future use
01 010000	LOWPAN_BC0	- LOWPAN_BC0 broadcast
01 010001	reserved	- Reserved for future use
...	reserved	- Reserved for future use
01 111110	reserved	- Reserved for future use
01 111111	ESC	- Additional Dispatch byte follows
10 xxxxxx	MESH	- Mesh Header
11 000xxx	FRAG1	- Fragmentation Header (first)
11 001000	reserved	- Reserved for future use
...	reserved	- Reserved for future use
11 011111	reserved	- Reserved for future use
11 100xxx	FRAGN	- Fragmentation Header (subsequent)
11 101000	reserved	- Reserved for future use
...	reserved	- Reserved for future use
11 111111	reserved	- Reserved for future use

Figure 2: Dispatch Value Bit Pattern

RFC6282

Section 5.3 of [RFC4944] also defines how to fragment compressed IPv6 datagrams that do not fit within a single link frame. Section 5.3 of [RFC4944] defines the fragment header's datagram_size and datagram_offset values as the size and offset of the IPv6 datagram before compression. As a result, all fragment payload outside the first fragment must carry their respective portions of the IPv6 datagram before compression. This document does not change that requirement. When using the fragmentation mechanism described in Section 5.3 of [RFC4944], any header that cannot fit within the first fragment MUST NOT be compressed.

Problem statement

- Per-hop fragmentation and reassembly has 2 issues:
  - Latency:
    - Increases end-to-end latency as you need to wait for each fragment at each hop
  - Reliability:
    - Limited memory → limited number of buffers (1-2?) → packet dropped when new frag received and old not fully reassembled yet
    - No frag recovery: 1 frag loss == packet dropped
- Proposed solution:
  - Fragment forwarding:
    - Source fragments
    - Intermediate nodes relays
    - LBR reassembles

6LoWPAN: The Wireless Embedded Internet

a.k.a. “Carsten’s book”

2.5.2 L3 routing (“Route-Over”)

Layer-3 Route-Over forwarding is illustrated in Figure 2.6. In contrast to layer-2 mesh forwarding, layer-3 Route-Over forwarding does not require any special support from the adaptation layer format. Before the layer-3 forwarding engine sees the packet, the adaptation layer has done its work and decapsulated the packet – at least conceptually (implementations may be able to perform some optimizations by keeping the encapsulated form if they know how to rewrite it into the proper encapsulated form for the next layer-3 hop).

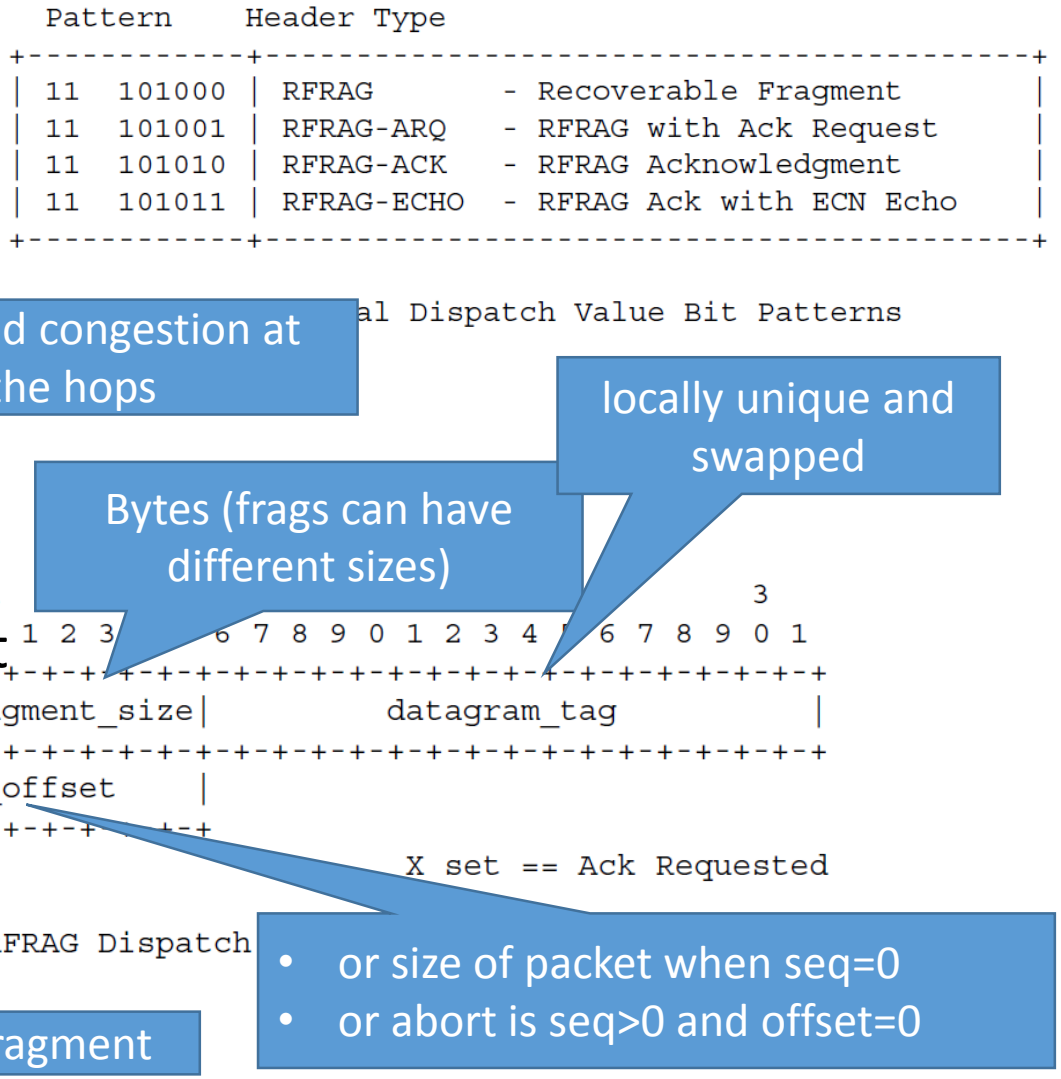
Note that this in particular means that fragmentation and reassembly are performed at each hop in Route-Over forwarding – it is hard to imagine otherwise, as the layer-3 addresses are part of the initial bytes of the IPv6 header, which is present only in the first fragment of a larger packet. Again, implementations may be able to optimize this process by keeping virtual reassembly buffers that remember just the IPv6 header including the relevant addresses (and the contents of any fragments that arrived out of order before the addresses).

→ being published as a combination of draft-bormann-lwig-6lowpan-virtual-reassembly-00 and draft-watteyne-6lo-minimal-fragment (WIP)

→ being simulated by Yatch on the 6TiSCH simulator

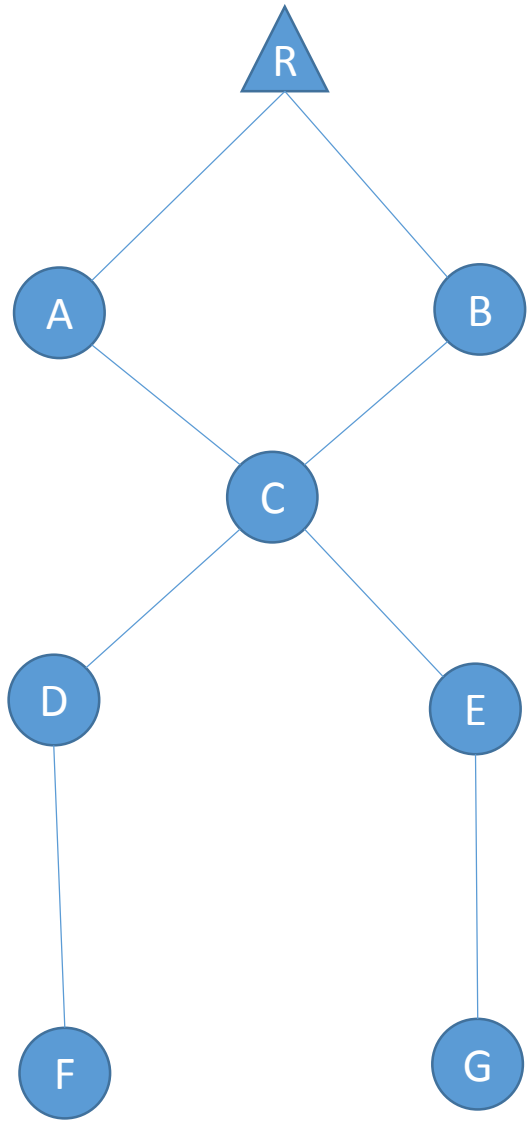
draft-thubert-6lo-forwarding-fragments-07

- Fragment forwarding
  - Locally unique label, swapped at each hop
- End-to-end ACK
  - ACK requested by source
    - for any fragment
  - ACK travels reverse LSP
- Fragment recovery
  - bitmap in RFRAG-ACK, one bit per fragment
- Flow control capabilities
- Different size per frag



6lo Frag. DT – Problem Statement (Proposal)

- Produce 2 documents (to be submitted to 6lo WG):
  - informational document
    - summarize fragmentation as standardized now
    - describes Carsten's virtual reassembly buffer implementation
    - discusses its limits
  - standards-track document
    - builds upon the first one
    - adds fragment recovery
    - *(can either be a replacement of draft-thubert-6lo-forwarding-fragments, or a rework)*
- Philosophy
  - keep activity as swift as possible
  - ideally close the DT after London
  - small DT, but regular information to WGs


R
A
B
C
D
E
F
G